Encyclopedia of Robotics
(will be inserted by the editor)

Visual Tracking

Eric Marchand

the date of receipt and acceptance should be inserted later

1 Synonyms

Visual localization

2 Definition

Visual tracking is a state estimation issue. From image measurements one has to consistently estimate the state of one or more objects over the discrete time steps in a video. Various measurements can be considered: pixel intensity (raw data), color, visual features (edges, lines, keypoints, motion field), etc. On the other side the state to be estimated can be 2D coordinates (center of gravity of the object), geometrical features (line, ellipse, etc.), bounding box, 3D rigid pose, homography, pose and scene structure (vSLAM), etc.

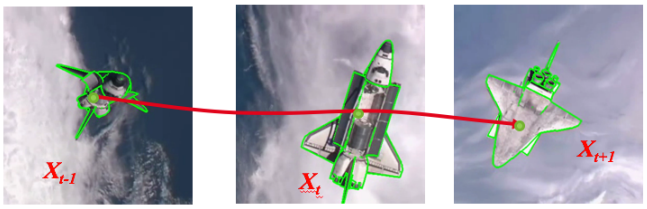


Fig. 1 Visual tracking has to consistently estimate the state (e.g., position $\mathbf{X}$) over time of an object in an image sequence.

Eric Marchand
Univ Rennes, Inria, CNRS, Irisa, France
E-mail: Eric.Marchand@irisa.fr

3 Overview

Elaboration of object tracking algorithms in image sequences is an important issue for researches and applications related to visual servoing, SLAM, visual odometry, and more generally for robot vision. A robust extraction and real-time spatio-temporal tracking process of visual cues is indeed one of the keys to success of a robot vision task.

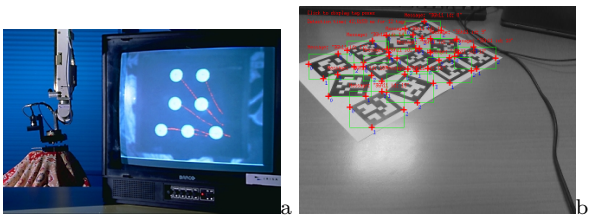
Until the early 2000s, almost all the vision-based registration and tracking techniques relied on markers or simple image processing techniques. Then various markerless methods quickly emerged in the literature. On one hand, markerless model-based tracking techniques improve clearly (but are in line with) marker-based methods. Meanwhile, template-based tracking methods arose from the motion estimation community. On the other hand, with the ability to easily match keypoints like SIFT, and the perfect knowledge of multi-view geometry, new approaches based on the estimation of the displacement of the camera arose. The late 2000s saw the introduction of keyframe-based Simultaneous Localization and Mapping (SLAM) that is a sequel of structure from motion approaches. Although vision-based tracking is still a difficult problem, mature solutions may now be proposed to the end-users and real-world or industrial applications can be foreseen (if not already seen). It is hopefully now possible to handle natural scenes featuring complex objects in various illumination conditions.

4 Key Research Findings

4.1 Visual tracking of low-level features

4.1.1 Fiducial marker detection and localization

In robotics, most early papers related to vision-based control considered very basic fiducial markers. The considered object is usually composed of "white dots on a black background" which are extracted and tracked using a simple connected-component analysis process. From a practical point of view, such algorithms are still useful to validate theoretical aspects of vision-based control research or for educational purposes. Furthermore, in some critical industrial processes such a simple approach ensures the required robustness (see Figure 2a).



a b

Fig. 2 Fiducial marker (a) Visual servoing using fiducial markers for a grasping task (b) April tag

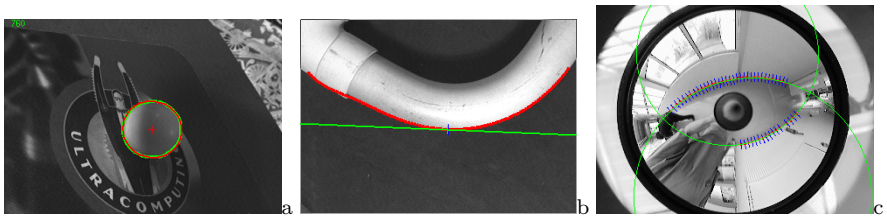


Fig. 3 Tracking 2D geometric features: tracking a sphere (a), a curve (b), and two ellipses (c).

More complex fiducial markers (such as April Tag (Olson 2011), see Figure 2b) allow achieving simultaneously both target identification and camera localization. To simplify the detection process and the underlying image processing algorithm, their design is ultimately simplified. Square shape and binary color combination are usually considered. Rectangle shape is first searched in a binarized image, and then camera pose with respect to the rectangle is computed (see Section 4.3.1).

4.1.2 Tracking contour-based 2D features

To track 2D geometric features (lines, circles, curves, see Figure 3), it is necessary to consider at the low level a generic framework that allows local tracking of edge points. Usually, the contours are sampled at a regular distance and, at these sample points, a one-dimensional search is performed to the normal of the contour for corresponding edges. This is implemented with convolution efficiency for real-time performance (e.g., Marchand and Chaumette (2005)). It is then possible to perform a robust linear estimation of the feature parameters using an Iteratively Reweighted Least Squares approach or RANSAC.

4.1.3 Keypoints or tracking by matching

Following the introduction of SIFT (Lowe 2004), the development of keypoints matching methodologies in the late 1990s allowed vision-based robotics reaching a new maturity level. The common framework for 2D keypoint matching usually considers three steps: keypoints extraction, description, and matching.

Keypoints extraction. Local features are extracted according to image properties computed from texture such as ‘cornerness”. Historically, Harris detector is a widely used corner detector that computes the cornerness score of each pixel from gradients of an image patch. The cornerness score allows classification into flat, edge and corner according to the intensity structure of the patch. FAST (Rosten et al. 2010) considers only pixels on a circle for fast extraction. To deal with scale issue, several scale-invariant detectors based on scale space theory have been proposed. Generally, a linear Gaussian scale space is built and local extrema on this space is selected as a keypoint. One of the first scale-invariant keypoint detector used Laplacian of Gaussian (LoG). For efficiency issue, LoG is approximated by a difference of Gaussian in SIFT detector (Lowe 2004). In SURF (Bay et al. 2008), the determinant of the Hessian is used as another scale-space operator and is computed efficiently with integral images.

Keypoints description The next step consists in computing a feature vector that fully describes the keypoint and its local neighborhood. The resulting descriptor should be made invariant to geometric and photometric variations. Rotation invariance is usually achieved after computing the orientation of the keypoint. Several ways exist such as using the peak of histogram of gradient in the keypoint neighborhood (Lowe 2004). SIFT descriptor (Lowe 2004) is based on histogram of oriented gradients (HOG). A similar framework is used in SURF (Bay et al. 2008).

Intensity comparisons-based approaches have recently been considered. In BRIEF (Calonder et al. 2012), a descriptor is composed of a binary string in which each binary digit is computed from intensity comparison between pairwise pixels. The descriptor is then composed of a binary string concatenating the result of a set of binary tests. This means that a binary descriptor is directly computed from the image patch while gradient-based approaches need additional computations. They are far more computationally efficient. To increase the discriminative property of descriptors, different designs of intensity comparisons have been proposed in ORB (Rublee et al. 2011) (rotation invariance) and BRISK (Leutenegger et al. 2011) (scale and rotation invariance).

Matching Keypoints matching usually considers a nearest neighbor searching approach. The idea is basically to find the keypoint in the reference image with the closest descriptor. If a binary feature descriptor is considered, brute-force matching with hamming distance (XOR) is used because it can be efficiently implemented with common CPUs. Keypoint matching has been formulated as a classification problem (Lepetit and Fua 2006). In that case, the view set of a keypoint under affine transformation is compactly described and treated as one class. At run-time, statistical classification tools such as randomized trees (Lepetit and Fua 2006) or random forests are used for deciding to which class an extracted keypoint belongs. Enforcing geometrical constraints between keypoints can also be used to ease and assess matching (Lowe 2001).

4.2 Visual tracking as a 2D motion estimation problem

In this section, the tracking problem is viewed as a motion estimation issue. The goal is to estimate the 2D motion model undergone by the object between the acquisitions of two images using only 2D image information (keypoint coordinates, pixels intensities,...).

4.2.1 Motion estimation through points correspondences

If we now consider a 2D motion model noted w that transfers a point $\mathbf{x}_1$ in image I_1 to a point $\mathbf{x}_2$ in image I_2 according to a set $\mathbf{h}$ of parameters: $\mathbf{x}_2 = w(\mathbf{x}_1, \mathbf{h})$. $\mathbf{h}$ can account for a simple translation, sRt motion model (scale, rotation, translation), an affine motion model, a homography, etc. Let us note that, from a general point of view, there does not exist a 2D motion model or transfer function $w(.)$ that account for any 3D scene and any camera motion. Nevertheless, it can be demonstrated that, *when the scene is planar or when the camera undergoes a pure*

rotational motion, the coordinates of any corresponding points are linked thanks to a homography ${}^2\mathbf{H}_1$ such that

$$\mathbf{x}_2 = w(\mathbf{x}_1, \mathbf{h}) = {}^2\mathbf{H}_1\mathbf{x}_1 \tag{1}$$

Assuming that some keypoints can be matched (see previous section), a global motion model can be computed. The idea is to estimate the parameters $\mathbf{h}$ that define the motion model considering the minimization of a cost function defined by:

$$\widehat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} d(\mathbf{x}_{2i}, w(\mathbf{x}_{1i}, \mathbf{h}))^2 \tag{2}$$

which can usually be solved directly for $\mathbf{h}$. Distance $d(.,.)$ is usually the Euclidian distance and, in most cases (affine motion model, homography, etc.), equation (2) can be solved by using a Direct Linear Transform (DLT) algorithm (that is, a least square approach)

Fig. 4 Motion estimation from keypoints for vehicle platooning.

4.2.2 Motion estimation using direct image registration

The previous approaches consider pure geometric methods. An alternative is to fully embed the motion estimation in an image processing process. Appearance-based approaches, also known as template-based approaches, are different from the previous geometrical ones in the way that there is no low-level extraction nor matching processes (as presented in Section 4.1). In this case, the goal is to estimate the motion (or warp) between the current image and a reference template at the pixel intensity level.

Template registration. Let us consider that the appearance of the object is learned from the reference image I_0, the set of pixels $\mathbf{x}$ defines the template W to be tracked. We seek the new template location $w(\mathbf{x}, \mathbf{h}), \forall \mathbf{x} \in W$ in a new image I. As seen in Section 4.2.1, $\mathbf{h}$ are parameters of a motion model, usually a homography or an affine motion model. It is then possible to directly define this alignment or registration problem as the minimization of the dissimilarity (or maximization of the similarity) between the appearance in I_0 at the positions $\mathbf{x}$ in the region W and in I at the positions $w(\mathbf{x}, \mathbf{h})$. An analytic formulation of the registration problem can then be written as:

$$\widehat{\mathbf{h}} = \arg\min_{\mathbf{h}} \sum_{\mathbf{x} \in W} f\left(I_0(\mathbf{x}), I(w(\mathbf{x}, \mathbf{h}))\right) \tag{3}$$

where f is, here, a dissimilarity function. The choice of the similarity function is important. An obvious choice originated from the brightness constancy constraint stating that $I(w(\mathbf{x},\mathbf{h})) = I_0(\mathbf{x})$ is to consider the sum of squared differences (SSD). In this case, it leads typically to the KLT algorithm (Shi and Tomasi 1994) for small patches and a translational model and to (Hager and Belhumeur 1998; Baker and Matthews 2004; Benhimane and Malis 2004) for large template and an affine or homography motion model. The problem can be rewritten as:

$$\widehat{\mathbf{h}} = \arg\min_{\mathbf{h}} \sum_{\mathbf{x}\in W} \left(I_0(\mathbf{x}) - I(w(\mathbf{x},\mathbf{h}))\right)^2 \tag{4}$$

This is a non-linear optimization problem which can be efficiently solved by a Gauss-Newton method.

It has to be noted that the SSD is not effective in the case of illumination changes and occlusions. Several solutions have been proposed to add robustness toward these variations. The former solution is to consider an M-Estimator (Hager and Belhumeur 1998). The later deals with the choice of the (dis-)similarity function such as local zero-mean normalized cross correlation (ZNCC) (Irani and Anandan 1998) or mutual information (MI) (Dame and Marchand 2012).

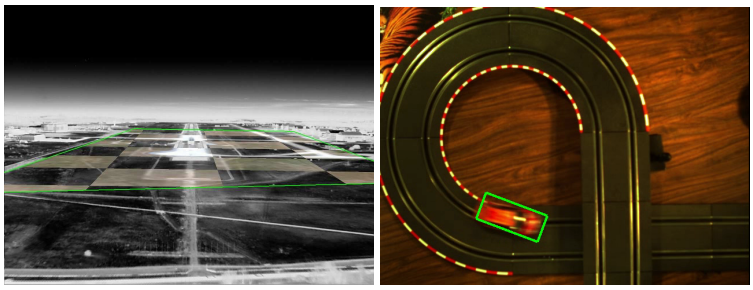


Fig. 5 Template-based tracking (a) runway tracking using MI-based Tracker (Dame and Marchand 2012) (b) Fast vehicle tracking using histogram-based correspondence (Comaniciu et al. 2000)

Template registration through histograms correspondence. Another approach considers distance between histograms. It has been extensively studied, especially since the successful application of the *mean shift* algorithm (Comaniciu et al. 2000). Let $\mathbf{q}_0(\mathbf{x}) = \{q^*_u(\mathbf{x})\}_{u=1\ldots m}$ denote the reference histogram, determined in the tracking initialization step. Then, tracking aims to find in each frame, the parameters $\mathbf{h}$ whose histogram $\mathbf{q}(w(\mathbf{x},\mathbf{h}))$ is the "closest" to the reference histogram $\mathbf{q}_0$. To achieve this, a correlation criterion in the histogram space is provided by Bhattacharyya coefficient:

$$\rho(\mathbf{x},\mathbf{h}) = \rho(\mathbf{q}_0(\mathbf{x}), \mathbf{q}(w(\mathbf{x},\mathbf{h}))) = \sum_{u=1}^{m} \sqrt{q^*_u(\mathbf{x}) q_u(w(\mathbf{x},\mathbf{h}))}. \tag{5}$$

At each instant, motion parameters $\mathbf{h}$ can be estimated by solving:

$$\widehat{\mathbf{h}} = \arg\min_{\mathbf{h}} \sqrt{1-\rho(\mathbf{x},\mathbf{h})}. \tag{6}$$

The well-known drawback of choosing histograms as a representation for the object is the loss of spatial information (Hager et al. 2004), making difficult to track more complex motions than a simple translation. In the past few years different methods have been proposed to tackle this issue, the main objective being to add some spatial information on the object to track while keeping the robustness of histogram descriptors. As for kernel-based methods, Hager et al. (2004) proposed a Newton-like framework for using a set of multiple kernels. The spatial configuration between them allows recovering high-dimensional motion parameters.

4.3 3D visual tracking

When a 3D model of the object is available or can be estimated on-line, visual tracking can be expressed as a pose estimation problem. The pose is estimated knowing the correspondences between 2D measurements in the images and 3D features of the model. We consider here that these 3D features and 2D measurements in the image are composed of a set of points.

Let us denote $\mathcal{F}_c$ the camera frame and ${}^c\mathbf{T}_w$ the transformation that fully defines the pose of $\mathcal{F}_w$ wrt. $\mathcal{F}_c$ (see Figure 6). ${}^c\mathbf{T}_w$ is a homogeneous matrix defined such that:

$$ {}^c\mathbf{T}_w = \begin{pmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{t}_w \\ \mathbf{0}_{3\times 1} & 1 \end{pmatrix} \tag{7} $$

where ${}^c\mathbf{R}_w$ and ${}^c\mathbf{t}_w$ are the rotation matrix and translation vector from $\mathcal{F}_c$ to $\mathcal{F}_w$.

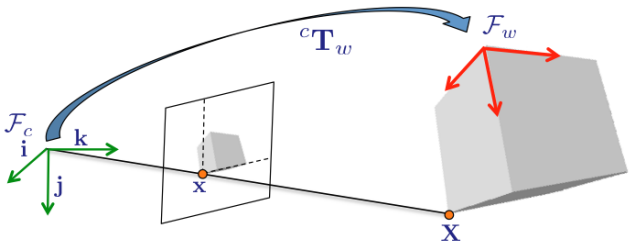


Fig. 6 Rigid transformation ${}^c\mathbf{T}_w$ between world frame $\mathcal{F}_w$ and camera frame $\mathcal{F}_c$ and perspective projection

Let us consider that the camera is calibrated and that the coordinates are expressed in the normalized space (Hartley and Zisserman 2001). If we have N points ${}^w\mathbf{X}_i, i = 1..N$ whose coordinates expressed in $\mathcal{F}_w$ are given by ${}^w\mathbf{X}_i = ({}^wX_i, {}^wY_i, {}^wZ_i, 1)^\top$, the projection $\mathbf{x}_i = (x_i, y_i, 1)^\top$ of these points in the image plane is then given by:

$$ \mathbf{x}_i = \mathbf{\Pi}\ {}^c\mathbf{T}_w\ {}^w\mathbf{X}_i. \tag{8} $$

where $\mathbf{\Pi}$ is the projection matrix. Knowing 2D-3D point correspondences, $\mathbf{x}_i$ and ${}^w\mathbf{X}_i$, pose estimation consists in solving the system given by the set of equations (8) for ${}^c\mathbf{T}_w$.

Pose estimation is mainly a single image problem. Nevertheless, when an image stream is considered, it requires the tracking of the-low level measurements (e.g.,

$\mathbf{x}_i$). This can be done using for example the approaches presented in Sections 4.1 and 4.2.

4.3.1 PnP: pose estimation from n point correspondences

As far as 2D-3D point correspondences are concerned, pose estimation is known as the Perspective from N Points (PnP) problem.

P3P. At least 3 point correspondences are necessary to compute the pose. P3P is an old problem for which many solutions have been proposed. Most of the P3P approaches rely on a 2 steps solution. First an estimation of the unknown depth ${}^cZ_i$ of each point (in the camera frame) is done thanks to constraints (law of cosines) given by the triangle $C\mathbf{X}_i\mathbf{X}_j$ for which the distance between $\mathbf{X}_i$ and $\mathbf{X}_j$ and the angle between the two directions $C\mathbf{X}_i$ and $C\mathbf{X}_j$ are respectively known and measured. The estimation of the points depth is usually done by solving a fourth order polynomial equation (Fischler and Bolles 1981; Quan and Lan 1999). Once the three points coordinates are known in the camera frame, the second step consists in estimating the rigid transformation ${}^c\mathbf{T}_w$ that maps the coordinates expressed in the camera frame to the coordinates expressed in the world frame (3D-3D registration, see Section 4.5). More recently, Kneip et al. (2011) proposed a novel closed-form solution that directly computes the pose ${}^c\mathbf{T}_w$.

PnP ($n \geq 4$). As for the P3P, one can consider multi-stage methods that estimate the coordinates of the 3D points or of virtual points, as in the EPnP (Lepetit et al. 2009), and then achieve a 3D-3D registration. Direct or one stage minimization approaches have also been proposed. PnP is intrinsically a non-linear problem; nevertheless, a solution relying on the resolution of a linear system can be considered. The Direct Linear Transform (DLT) is certainly the oldest one (Hartley and Zisserman 2001). It consists in solving the homogeneous linear system built from equations (8), for the 12 parameters of the matrix ${}^c\mathbf{T}_w$. Obviously and unfortunately, being over-parameterized, this solution is very sensitive to noise and a solution that explicitly takes the non-linear constraints of the system into account should be preferred. An alternative and very elegant solution has been proposed in Dementhon and Davis (1995): POSIT. Considering that the pose estimation problem is linear under the scaled orthographic projection model (weak perspective projection), Dementhon and Davis (1995) proposed to iteratively go back from the scaled orthographic projection model to the perspective one. An advantage of this approach is that it inherently enforces the non-linear constraints and is computationally cheap.

The "gold-standard" solution to the PnP consists in estimating the six independent parameters of the transformation ${}^c\mathbf{T}_w$ by minimizing the norm of the reprojection error using a non-linear minimization method such as a Levenberg-Marquardt (Hartley and Zisserman 2001; Marchand et al. 2016). Minimizing this reprojection error provides the Maximum Likelihood estimate when a Gaussian noise is assumed on measurements (i.e., on point coordinates $\mathbf{x}_i$). Another advantage of this approach is that it allows integrating easily the non-linear constraints induced by the PnP problem and provides an optimal solution. Denoting

$\mathbf{q} \in SE(3)$ a minimal representation of ${}^{c}\mathbf{T}_{w}$ ($\mathbf{q} = ({}^{c}\mathbf{t}_{w}, \theta\mathbf{u})^{\top}$ where θ and $\mathbf{u}$ are the angle and the axis of the rotation ${}^{c}\mathbf{R}_{w}$), the problem can be formulated as:

$$\widehat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} d\big(\mathbf{x}_i, \boldsymbol{\Pi}\, {}^{c}\mathbf{T}_{w}(\mathbf{q})\, {}^{w}\mathbf{X}_i\big)^2 \tag{9}$$

where $d(\mathbf{x}, \mathbf{x}')$ is the Euclidian distance between two points $\mathbf{x}$ and $\mathbf{x}'$. This method requires an initial value for $\mathbf{q}$ which can be provided by EPnP, POSIT or the DLT or when tracking is considered, by the pose obtained from the previous image.

In real life robotics applications, whatever the method chosen to solve the PnP, the solution must deal with the problem of robustness so as to account for noise, occlusion phenomena, changes in illumination, miss-tracking or errors in the correspondences and, more generally, for any perturbation that may be found in the video. A robust estimation process is usually incorporated into pose estimation. Voting techniques, Random Sample Consensus (RANSAC) (Fischler and Bolles 1981), M-Estimators, Least-Median of Squares (LMedS) have been widely used to solve this issue. RANSAC can also be considered to solve the initial 2D-3D matching issue.

4.3.2 Extension to markerless model-based tracking

Various authors have proposed different formulations of the pose estimation problem from other measurements than 2D points (Drummond and Cipolla 2002; Vacchetti et al. 2004; Comport et al. 2006; Choi and Christensen 2012; Petit et al. 2014). Although one can find some differences in these various solutions, the main idea is the following: as for equation (9) which is based on the distance between two points, the idea is to define a distance between a contour point in the image and the projected 3D contour underlying the corresponding 3D model.

Assuming an initial value of the pose is known (within a tracking process it is the pose estimated with the previous image), the 3D model is first projected into the image according to that pose. Contour $\mathbf{L}(\mathbf{q})$ is sampled (black points in Figure 7) and a search is performed along the edge normal to the contour (dashed lines) to find strong gradients in the next frame. Usually the point of maximum likelihood with respect to the initial sampled point $\mathbf{x}_i$ is selected from this exploration step. It is denoted by $\mathbf{x}_i$ in the following (white points in Figure 7).

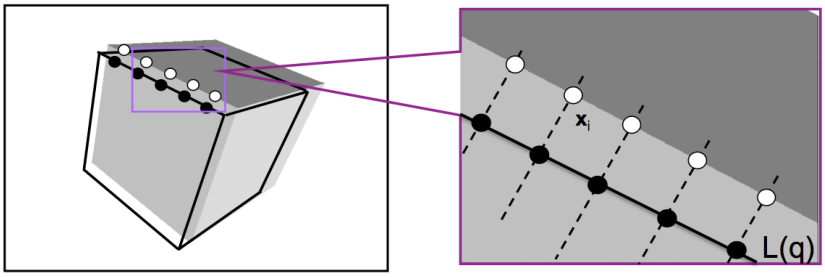
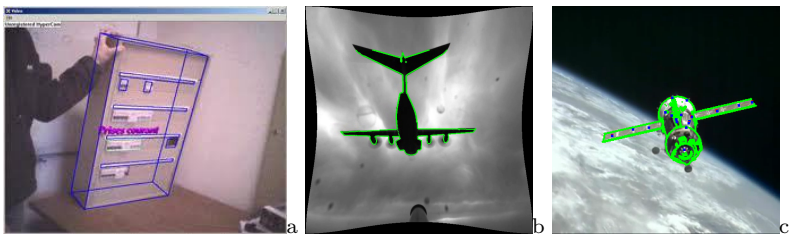


Fig. 7 Markerless model-based tracking: search for point correspondences between two frames and distance to be minimized.

A non-linear optimization is then used to estimate the camera pose which minimizes the errors between the selected points and the projected edges (Comport et al. 2006; Drummond and Cipolla 2002), that is:

$$\widehat{\mathbf{q}} = arg \min_{\mathbf{q}} \sum_{i} d_{\perp}(\mathbf{L}(\mathbf{q}), \mathbf{x}_i) \tag{10}$$

where $d_{\perp}(\mathbf{L}(\mathbf{q}), \mathbf{x}_i)$ is the squared distance between the point $\mathbf{x}_i$ and the projection of the contour of the model for the pose $\mathbf{q}$. This minimization is usually handled thanks to the Levenberg-Marquardt method. The main difference with respect to Section 4.3.1 is that a point-to-contour distance is considered rather than a point-to-point distance.



a b c

Fig. 8 Markerless model-based tracking (Comport et al. 2006; Petit et al. 2014).

The earliest approaches considered models composed with segments (see Figure 8a). More recent approaches proposed to render the 3D model (which can be arbitrarily complex) using a 3D rendering engine and a GPU (Petit et al. 2014; Choi and Christensen 2012). This allows automatically managing the projection of the model and determining visible and prominent edges from the rendered scene. An advantage of these techniques is to automatically handle the hidden faces removal process and to implicitly handle self-occlusions (see Figure 8b and 8c).

It has been noted that it could be interesting to merge 2D-3D registration methods along with 2D-2D ones. Most of the current approaches that integrate multiple cues in a tracking process are probabilistic techniques. Most of these approaches rely on the well-known Extended Kalman Filter or particle filter (e.g., (Kyrki and Kragic 2005)) but non-linear optimization techniques have also been considered (see Figure 8c). In Pressigout and Marchand (2007) the localization is based on both 2D-3D matching and a 2D-2D key-frame and temporal matching (which introduces multiple view spatio-temporal constraints in the tracking process). In Petit et al. (2014), color cues along with keypoints matching and edge-based model tracking are combined to provide a very robust tracker.

4.4 Pose from an a priori unknown model: vSLAM

Since a comprehensive or even a sparse 3D knowledge is not always easily available, the development of pose estimation methods that involve less constraining object model about the observed scene has been considered. The idea is to perform the estimation of the scene structure and the camera localization within the same

framework. This problem originally known as *structure from motion* was primarily handled off-line due to the high computational complexity of the solution. For real-time robotics, this leads to vSLAM (vision-based Simultaneous Localization And Mapping) that received much attention in the robotics community.

Considering monocular SLAM, two methodologies have been widely considered. The former is based on Bayesian filtering. Davison (2003) proposed to integrate data thanks to an Extended Kalman Filter whereas in Eade and Drummond (2006) (inspired from FastSLAM) a particle filter is considered. Within these approaches, measurements are sequentially integrated within the filter, updating the probability density associated with the state of the system (the camera pose, its velocity and the scene structure). All past poses being marginalized, the number of parameters to be estimated only grows with the size of the map. The latter approach is based on the minimization of reprojection errors (as in Section 4.3.1). It is known as bundle adjustment (BA) (Mouragnon et al. 2006; Klein and Murray 2007), which had proved to be very efficient and accurate in off-line applications. In Strasdat et al. (2010), it has been shown that, once the 'past" poses sequence has been sparsified (choosing adequately a reduced set of keyframes), the problem becomes tractable and BA proved to be superior to filter-based SLAM. Over the years, EKF-based vSLAM has been progressively replaced by keyframe and BA-based methods. Nowadays, real-time BA can operate on large-scale environment. It has to be noted that, when loop closure is not considered, vSLAM is closely related to visual odometry (VO) (Nistér et al. 2004; Scaramuzza and Fraundorfer 2011) which considers a local BA on a short sliding window.

Denoting $[\mathbf{q}]_M = (\mathbf{q}_1, \ldots, \mathbf{q}_t)$ a sequence of t camera poses (keyframes) and $[{}^w\mathbf{X}]_N = ({}^w\mathbf{X}_1, \ldots, {}^w\mathbf{X}_N)$ a set of N 3D points, the goal is, as for the PnP problem to minimize the error between the observations and the reprojection of 3D points. The error to be minimized is then given by:

$$([\widehat{\mathbf{q}}]_t,\ \widehat{[{}^w\mathbf{X}]_N}) = \arg \min_{([\mathbf{q}]_t, [{}^w\mathbf{X}]_N)} \sum_{j=1}^{t} \sum_{i=1}^{N} d\big(\mathbf{x}_{j_i}, \mathbf{\Pi}^j \mathbf{T}_w \, {}^w\mathbf{X}_i\big)^2$$

Initialization being an important issue, camera motion between a given keyframe and the current one is estimated using e.g. the five-point algorithm (Nistér 2004) and points are triangulated.

Mouragnon et al. (2006) and Mur-Artal et al. (2015) have clearly demonstrated the feasibility of a deterministic SLAM system for robotics. Nevertheless, such SLAM-based approaches lack absolute localization and are computationally expensive in large environments. To achieve real-time requirement and to cope with scale factor and the lack of absolute positioning issues, it has been proposed to decouple the localization and the mapping step. Mapping is handled by a full-scale BA or a keyframe-based BA. It is processed to fix the scale factor and to define the reference frame. Then, only a pose estimation (PnP) is performed on-line providing an absolute and reliable pose to the end-user. Such an approach has been successfully used for vehicle localization (e.g., (Royer et al. 2007)).

In BA-based vSLAM, only few pixels contribute to the pose and structure estimation process. As in Section 4.2.2, dense approaches such as DTAM (Newcombe et al. 2011b) allow each pixel contributing to the registration process (optimization is performed directly over image pixel intensities). This is also the case for LSD-SLAM (Engel et al. 2014). This latter approach is a keyframe method that

builds a semi-dense map, which provides far more information about the scene than feature-based approaches.

4.5 Pose in the 3D space

So far we considered a 2D-3D registration process. With some devices (e.g., multiple cameras systems) it is possible to get directly the 3D coordinates of the observed points. In this case, the registration can be done directly in the 3D space. The observed point ${}^1\mathbf{X}$ has to be registered with the model point ${}^2\mathbf{X}$ up to the transformation ${}^1\mathbf{T}_2$ that needs to be estimated.

Denoting $\mathbf{q} \in SE(3)$ a minimal representation of ${}^1\mathbf{T}_2$, the problem can be formulated as:

$$\widehat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} ({}^1\mathbf{X}_i - {}^1\mathbf{T}_2 {}^2\mathbf{X}_i)^2 \tag{11}$$

and solved using closed form solutions or robust Levenberg-Marquardt approaches. This is a simple problem when the matching between ${}^1\mathbf{X}_i$ and ${}^2\mathbf{X}_i$ is known (even with some outliers). When this matching is unknown, Iterative Closest Point ICP (Besl and McKay 1992) is an attractive solution.

In late 2010 new sensors (*kinect*, Intel RealSense, etc) have been introduced. These sensors provide in real time a dense 3D representation of the environment. Prior to the introduction of these cheap sensors, only more expensive time-of-flight cameras, heavy structured light systems and stereovision cameras existed. Kinect integrates a structured light (infra-red) depth sensor able to provide depth map at 30Hz. KinectFusion (Newcombe et al. 2011a) was one of the first systems that enables scene reconstruction and consequently camera localization in real-time. The idea is to simultaneously localize the camera and fuse live dense depth data building a global model of the scene. Indeed, the camera pose can be estimated by aligning the depth map data onto the current model (Newcombe et al. 2011a). This can be done by a modified version of the ICP that allows obtaining fast dense correspondences using closest point approximation. A fast point-to-plane ICP is finally used to register the current dense 3D map with the global model.

5 Examples of Application

Visual tracking is used in many robotics applications. Here are some non-exhaustive examples.

Visual servoing. Elaboration of object tracking algorithms in image sequences has been an important issue for research and application related to visual servoing. The use of fiducial markers allowed the validation of theoretical aspects of visual servoing research in the early 90's. More complex tracking such as the ones presented in Section 4.3.2 are now widely considered (see Figure 9).

Autonomous vehicle localization and navigation. vSLAM approaches are nowadays classically considered for vehicles navigation (self-driving cars, unmanned aerial vehicles and autonomous underwater vehicles). It is a possible alternative to Lidar.

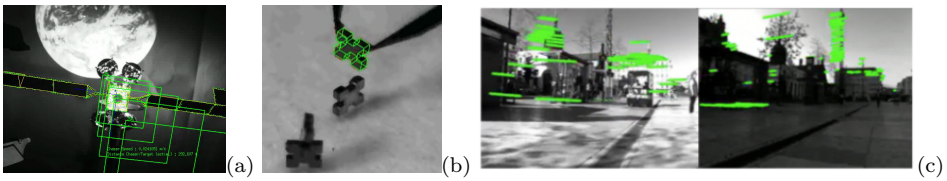


Fig. 9 Visual servoing for (a) space rendezvous (b) micro assembly (c) vision-based navigation

Space application. 3D tracking is a key requirement in space applications for autonomous uncooperative space rendezvous and proximity operations with space targets or debris (Petit et al. 2014) (see Figure 10).

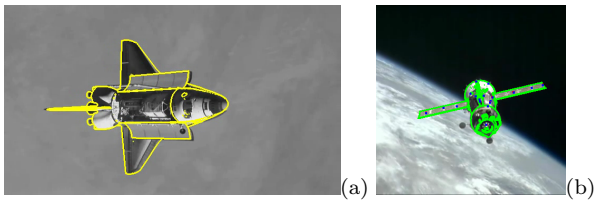


Fig. 10 Tracking for space rendezvous and proximity operations (Petit et al. 2014).

Mosaicing Image mosaics are a collection of overlapping images. The goal of the mosaicing problem is to find the transformation that relates the different images. Once the transformation between all the images is known, an image of the whole scene can be constructed. This can be handled using the mentioned motion estimation approaches or vSLAM. It has been widely used for underwater or aerial robotics contexts (see Figure 11).

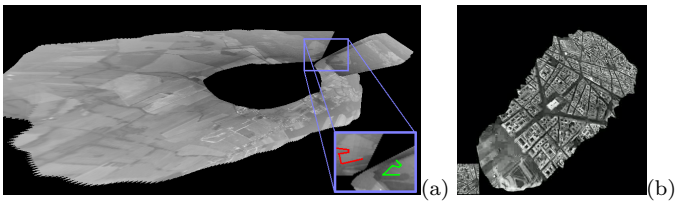


Fig. 11 Mosaicing for aerial operations (Dame and Marchand 2012)

Augmented reality Visual tracking is the basic tool for the development of augmented reality systems since a camera localization process has to be considered. Augmented reality is a key element of industry 4.0 with the next generation of sensor-based robots able to navigate and/or interact in complex unstructured environments together with human users. AR is important to inform the operator about the robot status.

6 Future Directions of Researches

Recently most tracking techniques have been revisited in the light of machine learning. This is the case, for example, for 2D tracking and motion estimation (DeTone et al. 2016; Wang et al. 2018), pose estimation and visual odometry (Kendall et al. 2015; Grabner et al. 2018), and 3D tracking (Byravan and Fox 2017). Most of these approaches consider Convolutional Neural Network and incorporate the tracking framework into an end-to-end deep learning paradigm. Although these new approaches based on deep regressors are not, to date, as precise as conventional ones, it is likely that they will become mainstream in the future.

7 Cross-references

- Visual SLAM
- Visual Odometry
- Visual Navigation
- Visual Servoing
- Video Mosaicing

References

Baker S, Matthews I (2004) Lucas-Kanade 20 years on: A unifying framework. Int Journal of Computer Vision 56(3):221–255

Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Computer Vision and Image Understanding 110(3):346–359

Benhimane S, Malis E (2004) Real-time image-based tracking of planes using efficient second-order minimization. In: IEEE/RSJ Int. Conf. on Intelligent Robots Systems, Sendai, Japan, pp 943–948

Besl P, McKay N (1992) A method for registration of 3-d shapes. IEEE Trans on Pattern Analysis and Machine Intelligence 14(2):239–256

Byravan A, Fox D (2017) Se3-nets: Learning rigid body motion using deep neural networks. In: IEEE Int. Conf. on Robotics and Automation,, pp 173–180

Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P (2012) BRIEF: Computing a local binary descriptor very fast. IEEE Trans on Pattern Analysis and Machine Intelligence 34(7):1281–1298

Choi C, Christensen H (2012) Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. Int Journal of Robotics Research 31(4):498–519

Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp 142–149

Comport A, Marchand E, Pressigout M, Chaumette F (2006) Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans on Visualization and Computer Graphics 12(4):615–628

Dame A, Marchand E (2012) Second order optimization of mutual information for real-time image registration. IEEE Trans on Image Processing 21(9):4190–4203

Davison A (2003) Real-time simultaneous localisation and mapping with a single camera. In: IEEE Int. Conf. on Computer Vision, pp 1403–1410

Dementhon D, Davis L (1995) Model-based object pose in 25 lines of codes. Int Journal of Computer Vision 15:123–141

DeTone D, Malisiewicz T, Rabinovich A (2016) Deep image homography estimation. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'16

Drummond T, Cipolla R (2002) Real-time visual tracking of complex structures. IEEE Trans on Pattern Analysis and Machine Intelligence 24(7):932–946

Eade E, Drummond T (2006) Scalable monocular slam. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'2006, vol 1, pp 469–476
Engel J, Schöps T, Cremers D (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision, ECCV'14
Fischler N, Bolles R (1981) Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communication of the ACM 24(6):381–395
Grabner A, Roth P, Lepetit V (2018) 3d pose estimation and 3d model retrieval for objects in the wild. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)
Hager G, Belhumeur P (1998) Efficient region tracking with parametric models of geometry and illumination. IEEE Trans on Pattern Analysis and Machine Intelligence 20(10):1025–1039
Hager G, Dewan M, Stewart C (2004) Multiple kernel tracking with SSD. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'04, pp 790–797
Hartley R, Zisserman A (2001) Multiple View Geometry in Computer Vision. Cambridge University Press
Irani M, Anandan P (1998) Robust multi-sensor image alignment. In: IEEE Int. Conf. on Computer Vision, ICCV'98, Bombay, India, pp 959–966
Kendall A, Grimes M, Cipolla R (2015) Posenet: A convolutional network for real-time 6-dof camera relocalization. IEEE International Conference on Computer Vision, ICCV pp 2938–2946
Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'07), Nara, Japan
Kneip L, Scaramuzza D, Siegwart R (2011) A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2011, pp 2969–2976
Kyrki V, Kragic D (2005) Integration of model-based and model-free cues for visual object tracking in 3d. In: IEEE Int. Conf. on Robotics and Automation, ICRA'05, Barcelona, Spain, pp 1566–1572
Lepetit V, Fua P (2006) Keypoint recognition using randomized trees. IEEE Trans on Pattern Analysis and Machine Intelligence 28(9):1465–1479
Lepetit V, Moreno-Noguer F, Fua P (2009) EPnP: An accurate O(n) solution to the PnP problem. Int Journal of Computer Vision 81(2):155–166
Leutenegger S, Chli M, Siegwart R (2011) BRISK: Binary robust invariant scalable keypoints. In: International Conference on Computer Vision, pp 2548–2555
Lowe D (2001) Local feature view clustering for 3d object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001
Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int Journal of Computer Vision 60(2):91–110
Marchand E, Chaumette F (2005) Feature tracking for visual servoing purposes. Robotics and Autonomous Systems 52(1):53–70, special issue on "Advances in Robot Vision", D. Kragic, H. Christensen (Eds.)
Marchand E, Uchiyama H, Spindler F (2016) Pose estimation for augmented reality: a hands-on survey. IEEE Trans on Visualization and Computer Graphics 22(12):2633–2651
Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2006) Real time localization and 3d reconstruction. In: IEEE Int. Conf. on Computer Vision, vol 1, pp 363–370
Mur-Artal R, Montiel J, Tardos J (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans on Robotics 31(5):1147–1163
Newcombe R, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohi P, Shotton J, Hodges S, Fitzgibbon A (2011a) Kinectfusion: Real-time dense surface mapping and tracking. In: IEEE/ACM Int. Symp. on Mixed and Augmented Reality, ISMAR'11, Basel, pp 127–136
Newcombe R, Lovegrove S, Davison A (2011b) DTAM: Dense tracking and mapping in real-time. In: IEEE Int. Conf. on Computer Vision, pp 2320–2327
Nistér D (2004) An efficient solution to the five-point relative pose problem. IEEE Trans on Pattern Analysis and Machine Intelligence 26(6):756–770
Nistér D, Naroditsky O, Bergen J (2004) Visual odometry. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition
Olson E (2011) Apriltag: A robust and flexible visual fiducial system. In: IEEE Int. Conf. on Robotics and Automation, ICRA'11, pp 3400–3407

Petit A, Marchand E, Kanani A (2014) Combining complementary edge, point and color cues in model-based tracking for highly dynamic scenes. In: IEEE Int. Conf. on Robotics and Automation, ICRA'14, Hong Kong, China, pp 4115–4120

Pressigout M, Marchand E (2007) Real-time hybrid tracking using edge and texture information. Int Journal of Robotics Research 26(7):689–713

Quan L, Lan Z (1999) Linear n-point camera pose determination. IEEE Trans on Pattern Analysis and Machine Intelligence 21(8):774–780

Rosten E, Porter R, Drummond T (2010) Faster and better: A machine learning approach to corner detection. IEEE Trans on Pattern Analysis and Machine Intelligence 32(1):105–119

Royer E, Lhuillier M, Dhome M, Lavest J (2007) Monocular vision for mobile robot localization and autonomous navigation. Int Journal of Computer Vision 74(3):237–260

Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: Int. Conf. on Computer Vision, pp 2564–2571

Scaramuzza D, Fraundorfer F (2011) Visual odometry. IEEE Robotics Automation Magazine 18(4):80–92

Shi J, Tomasi C (1994) Good features to track. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94, Seattle, Washington, pp 593–600

Strasdat H, Montiel J, Davison A (2010) Real-time monocular SLAM: Why filter? In: Int. Conf. on Robotics and Automation, ICRA'10, Anchorage, USA, pp 2657–2664

Vacchetti L, Lepetit V, Fua P (2004) Stable real-time 3d tracking using online and offline information. IEEE Trans on Pattern Analysis and Machine Intelligence 26(10):1385–1391

Wang C, Galoogahi HK, Lin C, Lucey S (2018) Deep-LK for efficient adaptive object tracking. In: IEEE Int. Conf. on Robotics and Automation (ICRA), pp 627–634